TRUBLE	ECIAS	SESION	4!	2.18.	2.5				4-6-4-6
18)									
	1								
waE:	pushl	/ebp	10-						
		/esp			#/eax	= 1			
	lua-0	17 (.1.0	471	1.000	# /e	~=i			
	5000	\$2.1	ecx		""#	Tex	41		
	leal	(ikeo	x,8)	, ·/·es	×	1/eax	si		
	subl	·/·eax	1.1.e	dx	1	fleax	-8:	F = j +	-:
	loal	(/ear	(, /· es	x, 4	·/eax	#/6	ax =	1+41=	Si
	more	ucod2	(./ec	x, 1/-e	x, 41/e	x #(a)	ration	2+41+	(Si 64)
	addl	watt	(-/.ec	x, /e	dx,4)/	eax #	Quo	44 4 4 5	+ (7:04)
	more	·/ebp	1.1.05	6			1,-		
+	popl	repp					+		
	ret		4-1-	+			++		
111		-							
		Colu			+-+-				
		7547					11		
@ wie	xt2 CN	ZEH3	34	1=2					
12									
9)-13									
c) 13									
								27 (28 2-1-	4-1-1-
01).6			1.00						Constant
					6 2.11	\d		5005	
e) 1	, , , , , , , ,	\$++	1	3.2	6-7-11-	5 7		794	d'Crocos
0,8			0,5		13				
87 1			1.	212	1 7 171	+ 1,6	6.7	17.7	ciclo
0 0	7		06	, ,, ,	5				
				<u> </u>					
	2/75 J	1,167	· -	13 h					
1	78F	+=					100		
							+++		
.19).0) b)								111.1
Sx	11	710		SZ					
-XX	1	1-14		134	table			1 4	134
					Co3				-40
	• • •							aux	404
								1 elop	
	i i i				tabla			@ ret	4-18
1.1.1.2					[99]		-	191	++12
	4. 6	1-1-1				-+4000		×	++16+
	2/2/km.	+36			n	-+4004		4	++20+
	13	1+40							
and I have a little		1 1 1 1 1 1 1 1 1				1 1 1		BE	110401

MS ACCIESTALS

c) return (*x+aux i3);	
. move 12 (/ebp), /eax	e) i=) × y; wore 44(/elp), /eax
more (/eax), /eax	wore (6(/etp)./ecx
addl-4(tebp), leax	hull leax, lecx
	in a College of the College
d). aux.id = F(& (*p1) tablat; 1, 4;),
pudu (6(/ebp)	1) aux. c2(i \ = aux c2 [23])
more 8(/ebp), /eax # P1	movb -13(/ebp), /al
worl -44 (/ebp):/ecx #	lend -40(/ebp)./ecx
addl / eax, / ecx #Hablatj	addl -48 (/ebp);/eex.
pushe /ecx	moub /al (/ecx)
case F	+2 + + + + + + + + + + + + + + + + + +
adal \$8,1/esp	
wore /eax, -40(/ebp)	
g) for (i=0, (i <y) &&="" (i<(*p1).n);="" i<="" td=""><td>=1+51</td></y)>	=1+51
(*p1). +abla (i) i1 = (*p1). +ab	CERTIS TIL
marl \$0, /eax #1=0	
word 8(:/ebp)//ecx #p1	
pushlifebx	
for augh 16 (:/ebp):/eax #;	44
ige lifer curpl 4000 (kex). /eax #	(+01).x
1 CAR JUNE	addl /ecx, /edx
mul \$40, leax leax 1	table [i] is #(*p1) table [i] is
mare 36(/edx), /ebx	table [i] is #(*p1) table [i] .id #:(*p1), table [i] .i3
36 0 \$40, /eax /edx \$ → more 36(/edx), /ebx 7 addl /eax, /ebx \$	table Ci] is #(*p1) table C] is 4 (*p1) table Ci] is 4 (*p1) table Ci] is + i
158 (360, /eax /edx # mare 36(/edx), /ebx + addl /eax, /ebx + more /ebx, (/edx) #	table [i] is #(*p1) table [c] is #: (*p1), table [ci] is * (*p1): table [ci] is + 1
158 (\$40, /eax /edx \$# → mare 36(·/edx), /ebx # addl /eax, /ebx # vnove /ebx (·/edx) # addl \$5, /eax #	table [i] is #(*p1) table [i] is 4 (*p1) table [i] is 4 (*p1) table [i] is + i
36 (1840, /eax /edx ≠ more 36(/edx), /ebx + addl /eax, /ebx + vnove /ebx, (·/edx) # addl \$5, /eax + jump for	table [i] is #(*p1) table [i] is # (*p1) table [i] is to (*p1) table [i] is to (*p1) table [i] is to (*p1) table [i] i
inul \$40, /eax /edx # inul \$40, /eax /edx # inul \$40, /ebx # addl /eax, /ebx # inul \$6(/edx), /ebx # inul \$5, /eax # jup for i)	table [i] is #(*p1) table [i] is #: (*p1). table [i] is # (*p1). table [i] is + 1 # i = i + 5
Jose 1840, leax ledx # more 36(/edx), lebx addl leax, lebx vnove lebx (ledx) # addl \$5, leax jump for i)	table [i] is #(*p1) table [i] is # (*p1) table [i] is * (*p1) table [i] is + 1
Jose (1840, /eax /edx ≠ 1 / 1 / 1 / 1 / 2 / 1 / 2 / 2 / 2 / 2 /	table [i] is #(*p1) table [i] is # (*p1). table [i] is * (*p1). table [i] is + i * i = i + 5 . i = 0 while (aux .cz [i] !=!/) {
Jose 1840, /eax /edx # more 36(/edx), /ebx addl /eax, /ebx vnove /ebx, (/edx) addl 35, /eax jump for i)	table [i] is #(*p1) table [i] is # (*p1). table [i] is # (*p1). table [i] is + i # i = i + 5 . i = 0 while (aux.cz [i] != !/) { aux. cz [i] != !/) { i + t
Jose (1840, /eax /edx # 1 / work 36(/edx), /ebx # addl /eax, /ebx # vnove /ebx (/edx) # addl \$5, /eax # jump for i) file popl /ebx W) if (aux.id !=4) aux.i3=i; else aux.i3=j;	table [i] is #(*p1) table [i] is # (*p1). table [i] is # (*p1). table [i] is # i = i + 5 . i = 0 . while (aux.c2 [i] != !!) . aux. (2 [i] = '#'; / y . ugue \$0, /cax
1000 \$40, /eax /edx # 1000 36(/edx), /ebx # addl /eax, /ebx # vnove /ebx, (/edx) # addl & 5, /eax # jump for 1) for popl /ebx 1) aux. 13 = 1; ese aux. 13 = 1; wore -40(/ebp), /eax	table [i] is #(*p1) table [i] is # (*p1). ta
1000 \$40, /eax /edx # 1000 36(/edx), /ebx # addl /eax, /ebx # vnove /ebx, (/edx) # addl & 5, /eax # i) addl & 5, /eax # i) aux.i3 = i; else aux.i3 = j; else aux.i3 = j; uove -40(/ebp), /eax uove -40(/ebp), /eax uove -40(/ebp), /eax	table [i] is #(*p1) table [i] is # (*p1). table [i] is / (*p1). table [i] is / i = 1 + 5 while (aux .cz [i] != !/) { aux. (2 [i] != !/) { aux. (2 [i] != !/) { while (aux .cz [i] != !/) { aux. (2 [i] != !/) { aux. (2 [i] != !/) { while (aux .cz [i] != !/) { aux. (2 [i] != !/)
1000 \$40, /eax /edx # 1000 \$40, /eax /ebx # 1000 4	table [i] is #(*p1) table [i] is # (*p1) table [i] is # (*p1) table [i] is # i = i + 5 . i = 0 while (aux.cz [i] != !!) { aux.cz [i] = '#'; / / i + + leal -40(/ebp), /ecx while: cupl \$'i', 4(/ecx, /ecx) je finalice
1000 \$40, /eax /edx # 1000 \$40, /edx # 1000	table [i] is #(*p1) table [i] is # (*p1). table [i] is / (*p1). table [i] is / i = 1 + 5 while (aux .cz [i] != !/) { aux. (2 [i] != !/) { aux. (2 [i] != !/) { while (aux .cz [i] != !/) { aux. (2 [i] != !/) { aux. (2 [i] != !/) { while (aux .cz [i] != !/) { aux. (2 [i] != !/)
1000 \$40, /eax /edx # 1000 36(/edx), /ebx 40 26x /ebx 40 26x /edx # 40 40 40 40 40 40 40	table [i] is #(*p1) table [i] is # (*p1) table [i] is # (*p1) table [i] is # i = i + 5 . i = 0 while (aux.cz [i] != !!) { aux.cz [i] = #'; / y unule (cupl \$'.', 4(/ecx,/eax)) je finalice unule \$#' 4(/ecx,/eax) incl /eax
inul \$40, /eax /edx # mare 36(/edx), /ebx addl /eax, /ebx move /ebx (/edx) # addl \$5, /eax jup for i) flor popl /ebx see aux. i3=j; ase aux. i3=j; uove -40(/ebp), /eax mare 36(/eax), /edx fupl /eax, 161/ebp) je else move -48(/ebp), /edx	table [i] is #(*p1) table [] is # (*p1) table [i] is * (*p1) table [i] is * i= i+ 5 . i=0 while (aux.c2 [i] != !!) { aux.c2 [i] = '#'; / y ugve \$C, /cax leal -40(/elp), /ecx while: cupl \$':', 4(/ecx,/eax) je finaline unco \$#' 4(/cx,/eax) incl /eax jup while
1000 \$40, /eax /edx # 1000 36(/edx), /ebx 40 26x /ebx 40 26x /edx # 40 40 40 40 40 40 40	table [i] is #(*p1) table [] is # (*p1) table [i] is * (*p1) table [i] is * i= i+ 5 . i=0 while (aux.cz [i] != !!) { aux.cz [i] = #"; / y ugue \$C, /cax leal -40(/elp), /ecx while: cupl \$:', 4(/ecx,/eax) je finaline unco \$#' 4(/cx,/eax) incl /eax jup while
inul \$40, /eax /edx # mare 36(/edx), /ebx addl /eax, /ebx move /ebx (/edx) # addl \$5, /eax jup for i) flor popl /ebx see aux. i3=j; ase aux. i3=j; uove -40(/ebp), /eax mare 36(/eax), /edx fupl /eax, 161/ebp) je else move -48(/ebp), /edx	table [i] is #(*p1) table [i] is # (*p1) table [i] is # (*p1) table [i] is # i = i + 5 . i = 0 while (aux.cz [i] != !/) { aux.cz [i] = #'; i++ leal -40(/ebp), /ecx while: cupl \$':', 4(/ecx,/eax) je finalice unce \$'#' 4(/ecx,/eax) incl /eax
inul \$40, /eax /edx # mare 36(/edx), /ebx addl /eax, /ebx move /ebx (/edx) # addl \$5, /eax jup for i) flor popl /ebx see aux. i3=j; ase aux. i3=j; uove -40(/ebp), /eax mare 36(/eax), /edx fupl /eax, 161/ebp) je else move -48(/ebp), /edx	table [i] is #(*p1) table [] is # (*p1) table [i] is * (*p1) table [i] is * i= i+ 5 . i=0 while (aux.cz [i] != !!) { aux.cz [i] = #"; / y ugue \$C, /cax leal -40(/elp), /ecx while: cupl \$:', 4(/ecx,/eax) je finaline unco \$#' 4(/cx,/eax) incl /eax jup while
inul \$40, /eax /edx # mare 36(/edx), /ebx addl /eax, /ebx move /ebx (/edx) # addl \$5, /eax jup for i) flor popl /ebx see aux. i3=j; ase aux. i3=j; uove -40(/ebp), /eax mare 36(/eax), /edx fupl /eax, 161/ebp) je else move -48(/ebp), /edx	table [i] is #(*p1) table [] is # (*p1) table [i] is * (*p1) table [i] is * i= i+ 5 . i=0 while (aux.cz [i] != !!) { aux.cz [i] = #"; / y ugue \$C, /cax leal -40(/elp), /ecx while: cupl \$:', 4(/ecx,/eax) je finaline unco \$#' 4(/cx,/eax) incl /eax jup while
inm(1 \$40, /eax /edx); mare 36(/edx), /ebx addl /eax, /ebx vnove /ebx, (/edx) # addl \$5, /eax jup for i) for popl /ebx h) if (aux.i1 !=4) aux.i3=i; else aux.i3=j; move -40(/ebp), /eax mare 36(/eax), /edx foupl :/eax, 161/ebp) je else move -48(/ebp), /edx	table [i] is #(*p1) table [] is # (*p1) table [i] is * (*p1) table [i] is * i= i+ 5 . i=0 while (aux.cz [i] != !!) { aux.cz [i] = #"; / y ugue \$C, /cax leal -40(/elp), /ecx while: cupl \$:', 4(/ecx,/eax) je finaline unco \$#' 4(/cx,/eax) incl /eax jup while